

[illegible][illegible]

FFFFFFFFF	AAAAAA	LL	DDDDDDDD	EEEEEEEEEE	FFFFFFFFF	
FFFFFFFFF	AAAAAA	LL	DDDDDDDD	EEEEEEEEEE	FFFFFFFFF	
FF	AA	AA	DD	DD	EE	FF
FF	AA	AA	DD	DD	EE	FF
FF	AA	AA	DD	DD	EE	FF
FF	AA	AA	DD	DD	EE	FF
FFFFFFFFF	AA	AA	DD	DD	EEEEEEEE	FFFFFFFFF
FFFFFFFFF	AA	AA	DD	DD	EEEEEEEE	FFFFFFFFF
FF	AAAAAAAAAA	LL	DD	DD	EE	FF
FF	AAAAAAAAAA	LL	DD	DD	EE	FF
FF	AA	AA	DD	DD	EE	FF
FF	AA	AA	DD	DD	EE	FF
FF	AA	AA	DD	DD	EE	FF
FF	AA	AA	DD	DD	EE	FF
		LLLLLLLLLL	DDDDDDDD	EEEEEEEEEE	FF	
		LLLLLLLLLL	DDDDDDDD	EEEEEEEEEE	FF	

MM	MM	DDDDDDDD	LL
MM	MM	DDDDDDDD	LL
MMM	MMM	DD	DD
MMM	MMM	DD	DD
MM	MM	DD	DD
MM	MM	DD	DD
MM	MM	DD	DD
MM	MM	DD	DD
MM	MM	DD	DD
MM	MM	DD	DD
MM	MM	DD	DD
MM	MM	DD	DD
MM	MM	DDDDDDDD	LLLLLLLLLL
MM	MM	DDDDDDDD	LLLLLLLLLL

.TITLE \$FALDEF - FAL CONTROL BLOCK DEFINITIONS
.IDENT 'V04-000'

COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
ALL RIGHTS RESERVED.

THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
TRANSFERRED.

THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
CORPORATION.

DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.

Facility: FAL (DECnet File Access Listener)

Abstract:

This module defines symbols for the FAL control blocks that are defined
by the \$FALWRKDEF and \$FALSTBDEF macros.

Environment:

The MDL translator must be used to convert FALDEF.MDL into
FALDEF.MAR (and FALDEF.B32).

Author: James A. Krycka, Creation Date: 16-JUN-1977

Modified By:

V03-007 JAK0145 J A Krycka 12-APR-1984
Track changes in DAP message building algorithm: rename
FALSQ_XMT to FALSQ_BLD, rename FALSQ_AUX to FALSQ_XMT, and
remove FALSQ_AUX_PTR.

V03-006 JAK0137 J A Krycka 12-MAR-1984
Add FALSQV_LOG_CNT logging option and FALSQV_DIS_PMR option.

V03-005 JAK0136 J A Krycka 07-MAR-1984
Expand the FAL work area to 8K bytes.
Extend size of Key Definition XAB storage area.

Replace FAL\$Q_NCB and FAL\$T_NCBBUF with FAL\$Q_FALLOG,
FAL\$T_FALLOG, FAL\$Q_SYSNET, and FAL\$T_SYSNET.
Add FAL\$T_PRTBUF1 and FAL\$T_PRTBUF2 for FAL logging.
Replace FAL\$S_LOGFLG with four subfields that identify FAL\$LOG
options by class.
Also add several new fields to store FAL\$LOG option values.
General cleanup and reorganization.

- V03-004 JAK0107 J A Krycka 06-MAY-1983
Extend size of Protection XAB storage area.
- V03-003 JAK0104 J A Krycka 17-MAR-1983
Add FAL\$Q_RMS and FAL\$S_RMS_PTR.
Add FAL\$V_DIS_RBK logging option.
Add FAL\$V_TERMINATE flag.
- V03-002 KRM0098 K Malik 06-Apr-1983
Add FAL\$V_NEWNAM flag for use during a rename operation to
tell FAL\$ENCODE_NAM and FAL\$LOG_RESNAM which NAM block to use.
- V03-001 KRM0070 K Malik 23-Nov-1982
Add FAL\$S_FAB2, FAL\$S_NAM2, FAL\$T_NAMESPEC, FAL\$T_EXPAND2,
and FAL\$T_RESULT2 for use by \$RENAME.
Increase size of arrays to support 32 Allocation and Key
Definition XABs.
Change FAL\$B_LOGFLG to FAL\$S_LOGFLG.
Rename some symbols for consistency.


```

:++
:$FALWRKDEF defines FAL work area offsets and symbols.
:--

```

```

$STRUCT FAL,WRKDEF

```

```

K RCVEFN,1
K XMTEFN,2
K MBXEFN,3
K MAXBUFSIZ,<64*512-1>
K MIN_RBK,1
K MAX_RBK,127
K DFLT_RBK,64
K DFLT_BPM,20
K DFLT_BPL,20

```

```

F FLG,0
S LOGGING,4,B
S ENABLE,5,B
S DISABLE,6,B
S MISCOPT,7,B
V <M

```

```

CNF_MSG
ATT_MSG
,6

```

```

FTM
BLK_IO
WILD

```

```

NEWNAM
TERMINATE
,3

```

```

RCVQIO
RCVAST
XMTQIO
XMTAST
MBXQIO
MBXAST
,2

```

```

LAST_MSG

```

```

RET_RFA
RET_RECNUM
RET_STV
,4

```

```

:-----
: Miscellaneous constants
:-----

```

```

: Link receive QIO event flag
: Link transmit QIO event flag
: Mailbox receive QIO event flag
: Maximum DAP buffer size supported
: Minimum RMS multi-block cache size in blocks
: Maximum RMS multi-block cache size in blocks
: Default RMS multi-block cache size in blocks
: Default number of bytes per DAP message to
:   display when DAP messages are logged
: Default number of bytes per line to display
:   when DAP messages are logged
:-----

```

```

:***** offset = ^X0000 = 0 *****
:-----

```

```

: FAL status flags follow (bits 00-63):
: Define bits 32-39 as FAL logging options
: Define bits 40-47 as FAL enable options
: Define bits 48-55 as FAL disable options
: Define bits 56-63 as FAL miscellaneous options

```

```

: DAP message state information:
:   Processed Configuration message
:   Processed Attributes message

```

```

: Miscellaneous state information:
:   File transfer mode (vs record access mode)
:   Block I/O mode (vs record I/O mode)
:   This is a wildcard operation (determined
:     via parse of filespec string)
:   Use NAM2 block (used during rename operation)
:   FAL has entered termination sequence

```

```

: QIO related state information:
:   Link receive QIO outstanding flag
:   Link receive AST delivered flag
:   Link transmit QIO outstanding flag
:   Link transmit AST delivered flag
:   Mailbox receive QIO outstanding flag
:   Mailbox receive AST delivered flag

```

```

: Routine call function modifiers:
:   (The next bit is input to FALSTRANSMIT)
:   Last DAP message in sequence to be blocked
:   (The next 3 bits are input to FAL$ENCODE_STS)
:   Include RFA field in Status message
:   Include RECNUM field in Status message
:   Include STV field in Status message
:

```


LOG_NAM
LOG_STA
LOG_MSG

LOG_AST

LOG_QIO

LOG_CNT
LOG_BIT6
.1

ENA_BIT0
.7

DIS_CRC
DIS_MBK
DIS_RBK

DIS_PMR

DIS_BIT4
.3

PARSE_ERR
USE_DBS
USE_SYS

USE_VER
USE_SC1

USE_SC2

.2
>

```

:
:
F STATE_CTX,Q
S TBL_ADDR,0,L
S CUR_ADDR,4,L
F VALUE,B
F RCVBUFIDX,B
F RBK_CACHE,B
F ,B,5
F QIOBUFSIZ,W
F DAPBUFSIZ,W
F LNKCHN,W
F MBXCHN,W
F RCVIOSB,Q
F XMTIOSB,Q
F MBXIOSB,Q
F MBX,Q

```

```

-----
: These flags specify FAL logging options:
: Enable logging of filename/function requests
: Enable logging of data thrupt statistics
: Enable logging of individual DAP messages
: as they are encoded and decoded
: Enable logging of DAP message packet AST
: completions
: Enable logging of DAP message packet QIO
: requests
: Enable logging of internal counters
: Spare

```

```

: These flags enable FAL features:
: Spare

```

```

: These flags disable FAL features:
: Disable file level CRC checksum computation
: Disable DAP message blocking
: Disable RMS multi-block caching to/from
: disk during block I/O file transfer mode
: Disable poor-man's routing (i.e., reject any
: filespec received that contains a nodename)
: Spare

```

```

: These flags signal a parsing failure or denote
: qualifiers which require additional processing
: Error in parse of FAL$LOG equivalence string
: Use specified DAP buffer size
: Use specified 2-byte operating system and
: file system type identification
: Use specified 4-byte DAP version number value
: Use specified 32-bit value as first part of
: system capabilities mask to send
: Use specified 32-bit value as second part of
: system capabilities mask to send

```

```

-----
: State transition table context:
: State table address
: Current table entry address
: State transition value
: Receive buffer index (into address table)
: Number of blocks in RMS multi-blocking cache
: Unused locations
: Largest QIO request that can be supported by
: FAL (i.e., buffer size available)
: Maximum DAP message size (determined via
: exchange of Configuration messages)
: Link channel #
: Associated mailbox channel #
: Link receive I/O status block
: Link transmit I/O status block
: Mailbox receive I/O status block
: Mailbox message descriptor block

```



```

F RCV,Q      : Receive DAP message descriptor
F XMT,Q      : Transmit DAP message descriptor
              : (used for blocking DAP messages)
F BLD,Q      : Build DAP message descriptor
              : (used in building a new DAP message)
              : (note that BLD buffer overlays XMT buffer)

F .L         : Spare
F RCVBUF,L,2 : Receive buffer address table
K RCVBUCNT,2 : Number of receive buffers
F RMS,Q      : RMS block buffer descriptor
F RMS_PTR,L  : Next byte pointer for RMS block buffer
F DISPLAY,W  : Mask of optional DAP messages to return
              : (copy of DAP$W_DISPLAY1 or DAP$W_DISPLAY2)
F RECEIVED,W : Mask of received XABs to link to XAB chain
              : (i.e., XABs generated as a result of
              : receiving optional DAP messages)

V <M         : Meaning:
  KEYXAB     : Key Definition XAB(s)
  ALLXAB     : Allocation XAB(s)
  DATXAB     : Date and Time XAB
  PROXAB     : Protection XAB
  RDTXAB     : Revision Date and Time XAB
  >

F ALLXABINI,L : Bit vector of Allocation XABs initialized
F KEYXABINI,L : Bit vector of Key Definition XABs initialized
F CHAIN_NXT,L : Address of last pointer cell in XAB chain
              : -----
              : ***** offset = ^X0080 = 128 *****
              : -----
              : (VOLNAME and DIRNAME must be contiguous)
F VOLNAME,Q   : Volume name descriptor (for 3-part name)
F DIRNAME,Q   : Directory name descriptor (for 3-part name)
F FALLOG,Q    : FAL$LOG equivalence string descriptor
F SYSNET,Q    : SYSSNET equivalence string descriptor (which
              : will be made into an NCB descriptor)
F USE_DBS,W   : User entered DAP buffer size
F USE_SYS,W   : User entered 2-byte operating system and file
              : system type identification field
F USE_VER,L   : User entered 4-byte DAP version number field
F USE_SC1,L   : User entered system capabilities mask (00-31)
F USE_SC2,L   : User entered system capabilities mask (31-63)
F .L,4        : Unused locations
F STB,L,16    : Statistics block
K STB,64      : Statistics block size
F DAP,T,192   : DAP control block storage area
K DAP,192     : DAP control block size
F .L,8        : Reserved for future expansion of DAP structure
F .L,5        : Unused locations
F DATATYPE,B  : Copy of DAP$B_DATATYPE
F ACCOPT,B    : Copy of DAP$B_ACCOPT
F ACCFUNC,B   : Copy of DAP$B_ACCFUNC
F RAC,B       : Copy of DAP$B_RAC
F FOP,L       : Copy of DAP$L_FOP
F NUMBER,L    : Requested random record number
              : -----
              : ***** offset = ^X0200 = 512 *****

```



```

F FAB,L,20
K FAB,80
F RAB,L,17
K RAB,68
F NAM,L,24
K NAM,96
F FHCXAB,L,11
K FHCXAB,44
F DATXAB,L,11
K DATXAB,44
F PROXAB,L,22
K PROXAB,88
F SUMXAB,L,3
K SUMXAB,12
F RDTXAB,L,5
K RDTXAB,20
F ,L,12
F TEMP,L
F TEMP,Q

```

```

F FILESPEC,T,256
K FILESPEC,255
F EXPAND,T,256
K EXPAND,255
F RESULT,T,256
K RESULT,255
F KEYBUF,T,256
K KEYBUF,255

```

```

F FAB2,L,20
K FAB2,80
F NAM2,L,24
K NAM2,96
F ,L,20
F FILESPEC2,T,256
K FILESPEC2,255
F EXPAND2,T,256
K EXPAND2,255
F RESULT2,T,256
K RESULT2,255

```

```

F ALLXAB,L,8
K ALLXAB,32
K MAX AID,31
F ,L,248

```

```

-----
: (FAB, RAB, NAM, and FHCXAB must be contiguous)
: FAB storage area
: FAB size
: RAB storage area
: RAB size
: NAM block storage area
: NAM control block size
: File Header Characteristics XAB storage area
: File Header Characteristics XAB size
: Date and Time XAB storage area
: Date and Time XAB size
: Protection XAB storage area
: Protection XAB size
: Summary XAB storage area
: Summary XAB size
: Revision Date and Time XAB storage area
: Revision Date and Time XAB size
: Unused locations
: Temporary longword work area
: Temporary quadword work area
:
: ***** offset = ^X0400 = 1024 *****
:
: File specification string buffer
: File specification string buffer size
: Expanded name string buffer
: Expanded name string buffer size
: Resultant name string buffer
: Resultant name string buffer size
: Key buffer
: Key buffer size
:
: ***** offset = ^X0800 = 2048 *****
:
: RMS structures for use by RENAME operation
: (FAB2 and NAM2 must be contiguous)
: FAB storage area
: FAB size
: NAM storage area
: NAM size
: Unsued locations
: File specification string buffer
: File specification string buffer size
: Expanded name string buffer
: Expanded name string buffer size
: Resultant name string buffer
: Resultant name string buffer size
:
: ***** offset = ^X0C00 = 3072 *****
:
: Allocation XAB storage area
: Allocation XAB size
: Largest area ID value supported by FAL
: Space for 31 other Allocation XABs in array
:
: -----

```



```

F KEYXAB,L,19
K KEYXAB,76
K MAX REF,31
F ,L,589
F MBXBUF,T,64
K MBXBUF,64
K MBXQUOTA,<64*2>
F ,L,16

```

```

F PRTBUF1,T,256
F PRTBUF2,T,256
K PRTBUF,256

```

```

M 1

```

```

F KEYNAM,L,8
K KEYNAM,32
F ,L,248

```

```

P 1

```

```

F FALLOG,T,256
K FALLOG,255
F SYSNET,T,256
K SYSNET,255
F VOLNAME,T,256
K VOLNAME,255
F DIRNAME,T,256
K DIRNAME,255

```

```

L WRKBLN
E

```

```

: ***** offset = ^X1000 = 4096 *****
: -----
: Key Definition XAB storage area
: Key Definition XAB size
: Largest key of reference value supported here
: Space for 31 alternate Key Definition XABs
: Mailbox buffer
: Mailbox buffer size
: Mailbox quota
: Unused locations
: -----
: ***** offset = ^X1A00 = 6656 *****
: -----
: Primary (non-AST-level) print buffer
: Secondary (AST-level) print buffer
: Print buffer size (for FAL logging)
: -----
: ***** offset = ^X1C00 = 7168 *****
: -----
: Key Name buffer
: Key Name buffer size
: Space for 31 alternate Key Name buffers
: -----
: ***** offset = ^X1C00 = 7168 *****
: -----
: Logical name translation buffer for FAL$LOG
: Logical name translation buffer size
: Logical name translation buffer for SYSSNET
: Logical name translation buffer size
: Volume name string buffer (to store
: (concatenated node specs plus device name)
: Volume name string buffer size
: Directory name string buffer
: Directory name string buffer size
: -----
: ***** offset = ^X2000 = 8192 *****
: -----
: Define length of this control block
:

```



```
;++  
;$FALSTBDEF defines FAL statistics block offsets and symbols.  
:--
```

```
$STRUCT FAL,STBDEF  
F RCV_PKT,L      : Total # DAP message packets received  
F RCV_MSG,L      : Total # DAP messages received  
F RCV_DAT,L      : Total # records/blocks received  
                  : (i.e., # DAP Data messages)  
F RCV_USR,L      : Total # bytes of user data received  
                  : (i.e., # bytes in all records/blocks)  
F RCV_LNK,L      : Total # bytes of link data received  
                  : (i.e., # bytes in all DAP messages)  
F XMT_PKT,L      : Total # DAP message packets transmitted  
F XMT_MSG,L      : Total # DAP messages transmitted  
F XMT_DAT,L      : Total # records/blocks transmitted  
                  : (i.e., # DAP Data messages)  
F XMT_USR,L      : Total # bytes of user data transmitted  
                  : (i.e., # bytes in all records/blocks)  
F XMT_LNK,L      : Total # bytes of link data transmitted  
                  : (i.e., # bytes in all DAP messages)  
F ,L,6           : Unused locations  
L STBBLN         : Define length of this control block  
E  
  
: End of module
```


DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY